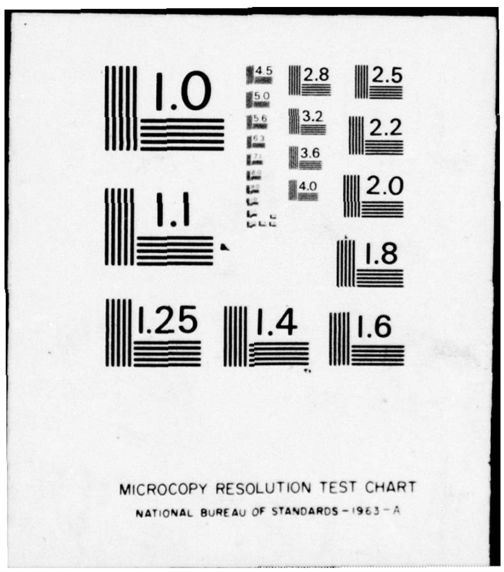
AD-A034 536 NAVAL RESEARCH LAB WASHINGTON D C F/G 15/4
A GRAPHICS PROGRAM FOR UPDATING THE CONFIDENCE REGION OF A TARG--ETC(U)
NOV 76 A SCHULTZ
UNCLASSIFIED NRL-MR-3415 NL

1 OF 1
AD
A034536

END
DATE
FILMED
2-77


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

NRL Memorandum Report 3415

(12)
NW

A Graphics Program for Updating the Confidence Region of a Target

ABRAHAM SCHULTZ

Systems Research Branch
Space Systems Division

November 1976

NAVAL RESEARCH LABORATORY
Washington, D.C.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Memorandum Report 3415	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A GRAPHICS PROGRAM FOR UPDATING THE CONFIDENCE REGION OF A TARGET		5. TYPE OF REPORT & PERIOD COVERED Final report on one phase of a continuing NRL problem.
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Abraham Schultz		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, D.C. 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NRL Problem B01-10 Project RR014-02-41-6152
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Research Laboratory Washington, D.C. 20375		12. REPORT DATE November 1976
		13. NUMBER OF PAGES 32
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

(6) (9) (10) (12) 33p. (11) (14) NRL-MR-3415

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

(16) RR01402 (17) RR0140241

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Ocean surveillance
Ship tracking
Man-machine systems

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A graphics program for updating the confidence region of a surveillance target subject to specified velocity constraints has been developed. The program is designed to operate in an interactive mode at a graphics terminal interface of an ocean surveillance information processing system. The initial confidence region of the target is assumed to be provided by a sensor contact report. The constraint set of velocities is specified in terms of upper and lower bounds on speed and heading and is either provided by a contact report or is selected by the ocean surveillance analyst. The

(Continues)

251950 JB

20. Abstract (Continued)

program displays the original and updated confidence regions. This program can be implemented on a graphics interactive system driven by a minicomputer.

CONTENTS

A GRAPHICS PROGRAM FOR UPDATING
THE CONFIDENCE REGION OF A TARGET

INTRODUCTION

The capability to update the confidence region associated with a target has many tactical applications. Useful analytic solutions to this problem are very difficult to obtain (see for example [1] and [2]). The graphics program presented here performs the dynamic update under fairly general conditions. The algorithms employed are mathematically rigorous; the main approximations made are associated with replacing the continuous boundary curve of a region with a discrete set of equally spaced points on the curve.

The operations of the Tenth Fleet during World War II has shown that information collected and processed by a wide area ocean surveillance system can have significant tactical applications. Human judgment plays a key role in the processing of tactical ocean surveillance data. A graphics terminal is a natural interface for a man-machine tactical information processing system. Computers have been used within wide area ocean surveillance systems for the direct recall of information or for elementary data correlation, but their capabilities have not been fully exploited in this application. A practical tactical information processing system should satisfy the following criteria:

Note: Manuscript submitted November 10, 1976.

(a) It should be able to process multi-source information.

(b) It should be an interactive system which can be used and understood by non mathematically oriented Navy analysts.

(c) It should enable the analyst to interrelate his understanding of the tactical situation with the performance characteristics of the sensor systems.

The confidence region update procedure which is described in this report satisfies the above requirements.

DESCRIPTION OF UPDAT PROGRAM

The program UPDAT, which updates the confidence region associated with a target is written in FORTRAN/PLOT 10 for the PDP-10 computer. It updates a confidence region associated with one or more sensor detects subject to specified velocity constraints. The three basic inputs required are:

(a) An initial confidence region P in position space which is assumed to contain the target.

(b) A constraint set V in velocity space.

(c) An update time T.

The UPDAT program computes the updated confidence region P_u in position space after an elapsed time T subject to the condition that the velocity of the target lies in the constraint set V. The boundary of the updated region is displayed on a graphics system.

The constraint set V is always defined by specifying the lower and upper bounds on speed (V1 and V2) and the lower and upper bounds on heading (A1 and A2). In general $V1 \leq V2$ and $A1 \leq A2$. Degenerate

cases in which V1 = V2 and A1 = A2 ($\mathcal{V}$ contains a single velocity vector, i.e., there is no uncertainty in velocity) or V1 = V2 and A1 < A2 ($\mathcal{V}$ contains all velocity vectors lying on a segment of a circle centered at the origin in velocity space) or A1 = A2, V1 < V2 ($\mathcal{V}$ contains all velocity vectors on a segment of a ray emerging from the origin in velocity space) are all permissible.

The analyst operating the UPDAT program has the following three options for the selection of an initial confidence region $\mathcal{P}$ in position space:

(a) Elliptical confidence region. The analyst inputs five parameters which define an arbitrary ellipse.

(b) An arbitrary convex confidence region. The analyst inputs the boundary points of this set.

(c) A wedge shaped confidence region. The analyst inputs four parameters which describes an arbitrary wedge.

OPERATIONAL INPUTS TO UPDAT

The operational inputs to UPDAT are either of the sensor performance type or the tactical type. The performance type of input arises from the known physical characteristics of the ocean surveillance (OS) sensors. An OS sensor measurement of a kinematic variable (e.g., such as speed, heading, position, line of bearing) yields an expected value of this variable and a confidence region (at a selected level) about the expected value. The tactical type of input arises from the analyst's understanding of the tactical context of the situation.

The sensor performance inputs arise from the following broad classes of sensors:

(a) Position measuring sensors.

(b) Line of bearing (LOB) measuring sensors.

(c) Velocity measuring sensors.

Elements within the first class include HF/DF and SOSUS. The output of this class of sensors is typically an elliptical confidence region. UPDAT requires the following five input parameters to describe an elliptical confidence region

(XO, YO, R1, R2, SO)

where

XO = longitude of center of ellipse,

YO = latitude of center of ellipse,

R1 = length of semi-major axis,

R2 = length of semi-minor axis (where $R_1 \geq R_2 > 0$), and

SO = orientation of semi-major axis relative to north.

The second class of inputs includes data from many of the passive sensors which measure LOB information. Elements of this class arise from isolated detects at individual HF/DF or SOSUS sites as well as from passive EW sensors. There are four input parameters which can be associated with a LOB message:

(XB, YB, B1, B2)

These parameters determine a wedge shaped confidence region as follows:

XB = longitude of sensor location

YB = latitude of sensor location

B1 = line of bearing of first great circle of wedge

B2 = line of bearing of second great circle of wedge, with B2 ≥ B1.

The coordinates of the vertex of the wedge are (XB, YB). Let BO denote

the observed bearing and let ΔB denote the maximum bearing error. Then

$$B1 = B0 - \Delta B$$
$$B2 = B0 + \Delta B$$

The determination of velocity from the observed Doppler shift of acoustic frequency is one source of velocity information. Velocity information can also arise from visual or radar detects. Intelligence reports (e.g., intercepted communications) provide another source of velocity data. In UPDAT the velocity data are assumed to be described by the four quantities:

$$(A1, A2, V1, V2)$$

where

A1 = lower bound of heading

A2 = upper bound of heading

V1 = lower bound of speed

V2 = upper bound of speed.

With

$A2 \geq A1$

$V2 \geq V1.$

Tactical considerations will often determine the constraint set $V = (A1, A2, V1, V2)$. Selected bounds (V1, V2) of the target's speed could originate from a combination of platform performance data and tactical context. As an example, the analyst should have available information on the top speed VMAX and the cavitation speed VC of different submarine classes. Under certain tactical situations V2 could equal VMAX while under other situations V2 will be equal to

VC. Selected bounds (A1, A2) of the target's heading could arise from a knowledge of the target's likely objectives or its observed behavior. The analyst can select $A1 = 0°$ and $A2 = 360°$ if there is no information to narrow down the heading estimates.

TACTICAL APPLICATIONS

Two applications of the UPDAT system at a central OS site are described here. In the first application, the analyst will use the UPDAT output to assist him in making track assignment decisions. In the second application, the UPDAT system is used to develop predictions of the locations of enemy combatants, which are then sent to search and attack units (SAUs). The track assignment application of the UPDAT system would develop with the following sequence:

(a) At time T_o an OS sensor has "detected" a target. This contact is at time T_o the last reported observation of the target. (The target's track could at time T_o consist of this single report.)

(b) At time $T_o + T$ a second detect was made. It is assumed that the signature information in the second detect is not by itself sufficient to assign it to the first target.

(c) Using the procedure outlined in the proceeding section a confidence region P in position space and a confidence region V in velocity space associated with the target at time T_o are developed. Let P' denote the elliptical confidence region associated with the new sensor detect at time $T_o + T$. Let P_u denote the update

to time $T_o + T$ of the set P subject to the velocity constraint set V. If P' doesn't intersect P_u, then the assignment of the second report to the existing track would not be consistent with the analyst's knowledge of the target's behavior. If P' does intersect P_u, the assignment is consistent with his knowledge but the analyst is not forced to make this decision.

These confidence regions are sketched in the figure shown below.

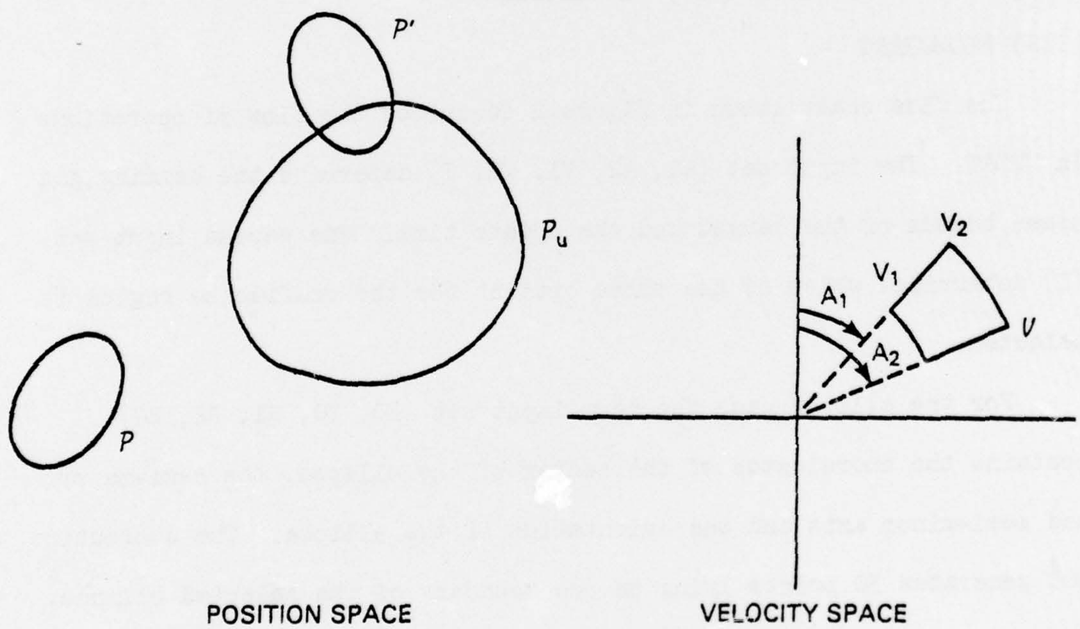


Fig. 1 — Elements of track assignment application

The tactical prediction application would proceed as follows: At time T_o a target detect was obtained. As discussed in the preceeding paragraph the confidence regions $\mathcal{P}$ and $\mathcal{V}$ are developed. SAUs are to attempt to acquire the target at time $T_o + T$. The UPDAT system can be used to combine the confidence regions $\mathcal{P}$ and $\mathcal{V}$ with the update time T to compute the updated confidence region $\mathcal{P}_u$ which is the region where the search effort should be concentrated. Given the shape and size of $\mathcal{P}_u$ and the sensor and platform characteristics of the SAUs, composite search strategies can be evaluated according to their associated total probability of detection.

UPDAT FLOWCHART

The flow chart shown in Figure 2 describes the flow of operations in UPDAT. The input set (A1, A2, V1, V2, T) determine the heading and speed bounds of the target and the update time. The second input set (I) determines which of the three options for the confidence region is selected.

For the ellipse case the next input set (XO, YO, R1, R2, SO) contains the coordinates of the center of the ellipse, the semi-major and semi-minor axis and the orientation of the ellipse. The subroutine BOE generates 30 points lying on the boundary of the selected ellipse. The call to CONDRAW plots this boundary on the graphics terminal. The subroutine TEST determines which subroutine (algorithm) is used to perform the update operation.

If the uncertainty in position space is zero ($R_1 = R_2 = 0$) or is small compared to T times the uncertainty in velocity space, then

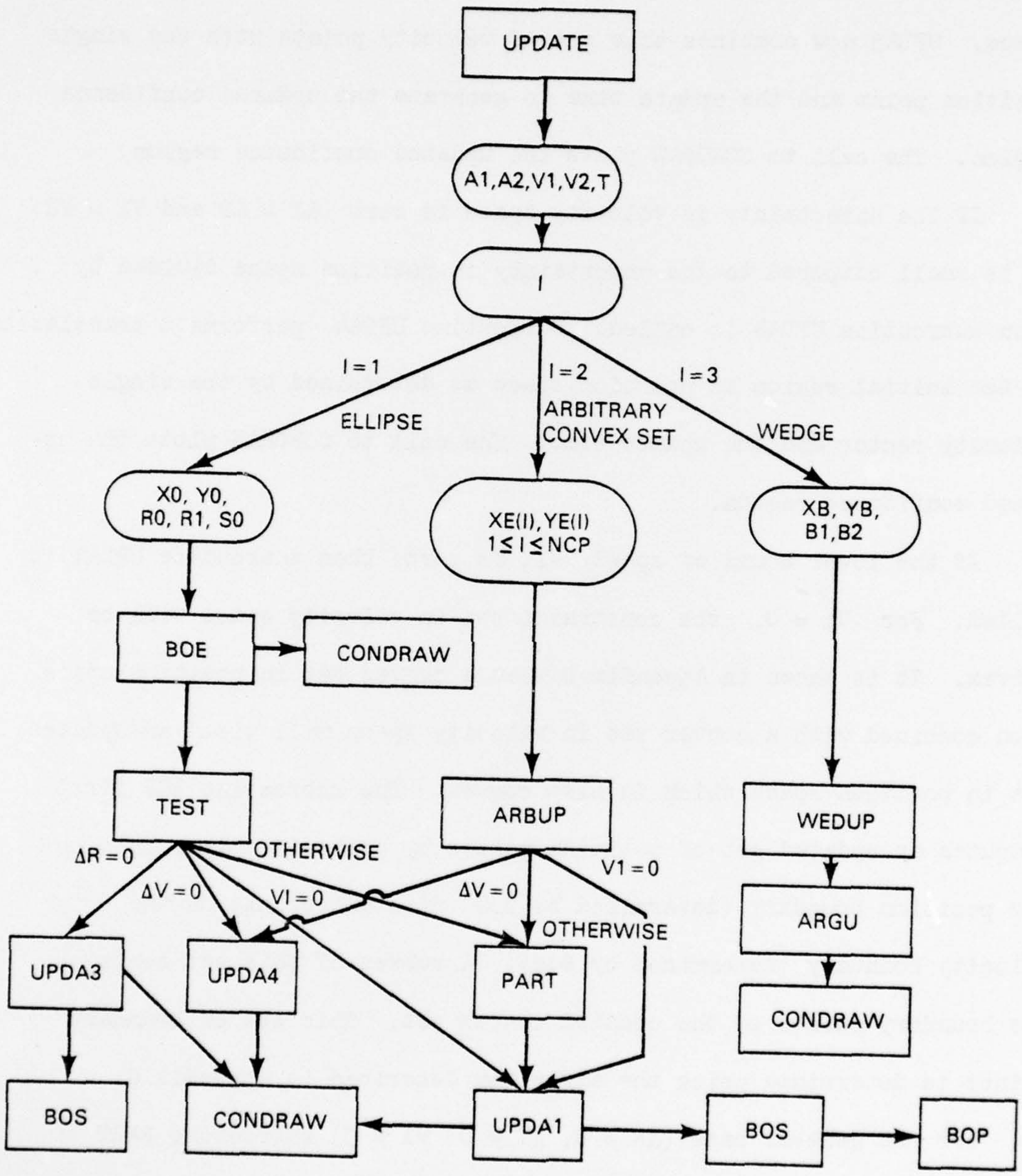


Fig. 2 — UPDAT flow chart

subroutine UPDA3 is called. This subroutine calls subroutine BOS which in turn generates the boundary points of the constraint set in velocity space. UPDA3 now combines this set of velocity points with the single position point and the update time to generate the updated confidence region. The call to CONDRAW plots the updated confidence region.

If the uncertainty in velocity space is zero (A1 = A2 and V1 = V2) or is small compared to the uncertainty in position space divided by T then subroutine UPDA4 is called. Subroutine UPDA4 performs a translation of the initial region in position space as determined by the single velocity vector and the update time. The call to CONDRAW plots the updated confidence region.

If the lower bound of speed, V1, is zero, then subroutine UPDA1 is called. For $V1 = 0$, the constraint set in velocity space will be convex. It is shown in Appendix B that a convex set in position space when combined with a convex set in velocity space will yield an updated set in position space which is also convex. The subroutine BOP first computes an updated set of position points by combining all points on the position boundary (determined by BOE) with all points on the velocity boundary (determined by BOS). A subset of this set contains the boundary points of the updated convex set. This set of boundary points is determined using the algorithm described in Appendix C.

For the general case ($\Delta R \neq 0$, $\Delta V \neq 0$, $V1 \neq 0$) subroutine PART is called. Subroutine PART decomposes V into a union of almost convex sets as determined by a preselected convexity criterion. For each one

of these nearly convex sets UPDA1 is called.

The case where I = 2 is very similar to the I = 1 case. Here the boundary points of the initial convex set are entered by the analyst. Subroutine ARBUP plays nearly the same role as subroutine TEST. The first call to CONDRAW plots the initial confidence region.

For the wedge case the next input set (XB, YB, B1, B2) contains the coordinates of the vertex of the wedge and the two angles which define the great circle boundaries of the wedge. Except for near the vertex point the updated confidence region will be bounded by two great circles. These great circles are determined by associating with the lines of bearings B1 and B2 two critical velocity vectors lying in V. The two critical velocities are determined exactly and hence except for near the vertex point WEDUP computes an exact update of the initial wedge. The first call to CONDRAW plots the initial confidence wedge and the second call to CONDRAW plots the updated confidence region.

EXECUTION OF UPDAT

This section describes how a Navy analyst would use UPDAT. The UPDAT program operates in an interactive mode. Data are entered on line on a graphics terminal with the program providing cues for the required input at each stage.

The system requests the first input parameter by printing out the phrase

HEIGHT:

The analyst then inputs the value of the height above sea level in nautical miles of the point of observation.

The system requests the second set of input parameters by

printing out the phrase:

LAT, LON:

The analyst then inputs the values of the latitude and longitude in (decimal) degrees of the point of observation.

The system requests the third set of input parameters by printing out the phrase:

INPUT A1, A2, V1, V2, T:

The analyst then inputs the five quantities

A1 = lower bound of heading of target (in degrees)

A2 = upper bound of heading of target (in degrees)

V1 = lower bound of speed of target (in knots)

V2 = upper bound of speed of target (in knots)

T = update time (in hours)

The heading bounds A1 and A2 are specified clockwise relative to due north. If the spread of uncertainty (A1, A2) in target heading does not span due north then $A1 \leq A2$. If the spread of headings does contain due north then $A1 > A2$.

The system now requests the fourth set of input parameters by printing out the phrase:

INPUT I: (I = 1 FOR ELLIPSE: I = 2 FOR ANALYST SELECTED SET:
I = 3 FOR WEDGE)

The analyst then inputs the value of I equal to 1, 2, or 3 depending on which option he selects for the initial confidence region of the target:

If the analyst inputs the value of I = 1, then the system requests the fifth set of input parameters by printing out the phrase

INPUT XO, YO, R1, R2, SO: (ELLIPSE PARAMETERS)

The analyst then inputs the five quantities

XO = longitude of center of ellipse (in degrees)

YO = latitude of center of ellipse (in degrees)

R1 = semi-major axis of ellipse (in nautical miles)

R2 = semi-minor axis of ellipse (in nautical miles)

SO = orientation of ellipse relative to due north (in degrees).

If the analyst inputs the value $I = 2$ then the system requests the fifth set of input parameters by printing out the phrase

INPUT N: (NUMBER OF POINTS IN BOUNDARY)

The analyst then inputs the number of points N $(N \leq 30)$ used to define the boundary of his selected convex confidence region. The system then requests the next set of input parameters by printing out the phrase

INPUT X, Y: (LONGITUDE AND LATITUDE OF BOUNDARY POINT)

The analyst then inputs the longitude and latitude (in degrees) of a boundary point of his selected set. This last operation will be automatically repeated N times.

If the analyst inputs the value $I = 3$, then the system requests the fifth set of input parameters by printing out the phrase:

INPUT XB, YB, B1, B2: (WEDGE PARAMETERS)

The analyst then inputs the four quantities

XB = longitude of vertex of wedge (in degrees)

YB = latitude of vertex of wedge (in degrees)

B1 = line of bearing of first great circle of wedge (in degrees)

B2 = line of bearing of second great circle of wedge (in degrees).

The lines of bearing are given relative to due north with $B1 \leq B2$,

unless the wedge spans due north.

After the parameters which define the initial confidence region have been entered, the system plots on the graphics terminal a 30 point approximation to P and then computes and displays a 30 point approximation to the boundary of updated confidence region P_u.

GRAPHIC OUTPUT

Two examples of UPDAT graphic output are described in this section. The basic program update operations are performed in spherical earth coordinates. The coordinate transformation which was used to map the spherical earth coordinates into the plane system of the graphic display is called a true view transformation and is one of the display options provided by the Graphic Analysis and Correlation Terminal (GACT) System.

Figure 3 and Figure 4 are examples of updates of an elliptical confidence region. In both cases the initial confidence region of the target is an ellipse with center (XO, YO) = (0, 0), semi-major axis of 50 nautical miles, and semi-minor axis of 25 nautical miles, and which is oriented at 45° relative to due north. In the first example (Figure 3) the target is assumed to have a heading which lies between 0° and 30° and a speed which lies between 20 and 30 knots.

In the second example, (Figure 4) the target is assumed to have a speed which lies between 20 and 30 knots, and a heading which is completely arbitrary. In both examples the update time is 8 hours. For both cases the constraint set in velocity space was partitioned into a disjoint union of almost convex sets and a separate update operation was performed using these sets.

The total updated confidence region is presented as a superposition

LAT, LON: 0 0

INPUT XO, YO, R1, R2, SO : 0 0 50 25 45

INPUT A1, A2, V1, V2, T : 0 30 20 30 8

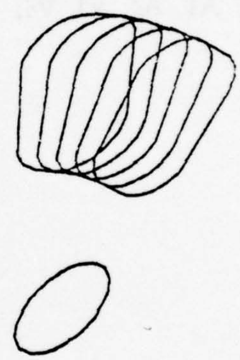


Fig. 3 — Example of updated confidence region, restricted heading bounds

LAT, LON: 0 0
INPUT X0, Y0, R1, R2, S0 : 0 0 50 25 45
INPUT A1, A2, V1, V2, T : 0 360 20 30 8

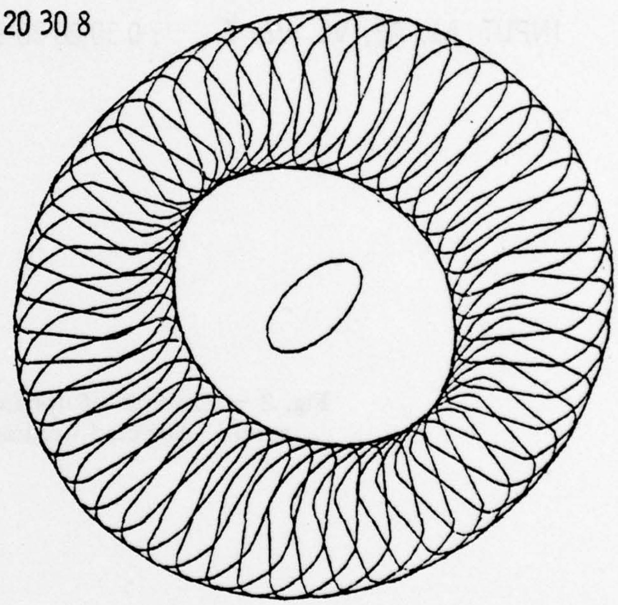


Fig. 4 — Example of updated confidence region, arbitrary heading

of adjacent updated confidence regions. It can be seen that the convexity criterion (see Appendix C) could have been relaxed without losing much definition in the updated confidence regions.

ACKNOWLEDGEMENTS

The author is grateful to Benny E. Martin who extended the FORTRAN version of UPDAT into a graphics program.

REFERENCES

1. Koopman, B. O., "Search and Screening," National Defense Research Committee, Washington, D. C., 1946.

2. Owens, M. E. B., "A Probability Density for the Future Position of a Vessel at Sea," Naval Research Laboratory Report 7128, August 26, 1970.

APPENDIX A

PROGRAM LISTING

This appendix contains a listing of the FORTRAN version of the UPDAT program. The subroutine CONDRAW used in the graphic version to generate a plot of the polygonal set associated with an array of points is not shown. The subroutine which computes the True View transformation from spherical earth coordinates to the plane graphic system is also not shown.

PROGRAM CONUP

```
      PROGRAM CON1(INPUT,OUTPUT,TAPE1)
      COMMON/BLOK1/XE(100),YE(100)
      COMMON/BLOK2/U(100),V(100)
      COMMON/BLOK3/XU(100),YU(100)
      COMMON/BLOK4/X0,Y0,R1,R2,S0,V1,V2,T,P
      COMMON/BLOK5/X1(3000),Y1(3000),Z(3000),TA(50)
      READ(1,) ND,X0,Y0,R1,R2,S0
      READ(1,) ND,A1,A2,V1,V2,T,P
      PRINT 10
 10 FORMAT(5X,26HBOUNDARY OF UPDATED REGION//)
     CALL TEST(A1,A2)
     STOP
     END
      SUBROUTINE TEST(A1,A2)
      COMMON/BLOK4/X0,Y0,R1,R2,S0,V1,V2,T,P
      DV1=V2-V1
      DV2=V2*(A2-A1)
      VU=AMAX1(DV1,DV2)
      VL=AMIN1(DV1,DV2)
      IF(V1.LE.0.01)  GO TO 10
      IF(R1/T.LE.0.05*VL) GO TO 20
      IF(VU.LE.0.05*R2/T) GO TO 30
      CALL PART(A1,A2,V1)
      RETURN
 10   CALL UPDA1(A1,A2)
      RETURN
 20   CALL UPDA3(A1,A2)
      RETURN
 30   CALL UPDA4(A1,A2)
      RETURN
      END
      SUBROUTINE PART(A1,A2,V1)
      E=0.04
      AL=2*ACOS(1.-E/V1)
      NT=INT((A2-A1)/AL)+1
      DAL=(A2-A1)/NT
      IF(NT.GT.1) GO TO 10
      CALL UPDA1(A1,A2)
      RETURN
 10   DO 50 I=1,NT
      A1=A1+DAL*(I-1)
      A2=A1+DAL
 50   CALL UPDA1(A1,A2)
      RETURN
      END
      SUBROUTINE UPDA1(A1,A2)
```

```
      COMMON/BLOK1/XE(100),YE(100)
       COMMON/BLOK2/U(100),V(100)
       COMMON/BLOK3/XU(100),YU(100)
       COMMON/BLOK4/X0,Y0,R1,R2,S0,V1,V2,T,P
       COMMON/BLOK5/X1(3000),Y1(3000),Z(3000),TA(50)
       CALL BOE(X0,Y0,R1,R2,P,S0,N)
       CALL  BOS(A1,A2,V1,V2,T,P,M)
       CALL BOP(T,M,N,L)
      DO 36 I=1,L
      PRINT 35,XU(I),YU(I)
   35 FORMAT(5X,F10.3,5X,F10.3)
   36 CONTINUE
       RETURN
      END
        SUBROUTINE BOE(X0,Y0,R1,R2,P,S0,N)
        COMMON/BLOK1/XE(100),YE(100)
        DS=P/R1
        N=INT(6.2832/DS)
        DO 10 I=1,N
       XB=R1*COS(I*DS)
       YB=R2*SIN(I*DS)
       XE(I)=X0+XB*COS(S0)-YB*SIN(S0)
   10  YE(I)=Y0+XB*SIN(S0)+YB*COS(S0)
        RETURN
        END
        SUBROUTINE BOS(A1,A2,V1,V2,T,P,M)
        COMMON/BLOK2/U(100),V(100)
        DV=P/T
        K1=INT((V2-V1)/DV)+1
        DV1=(V2-V1)/K1
        K2=INT(V2*(A2-A1)/DV)+1
        DA1=(A2-A1)/K2
       K3=K1
       DV3=DV1
       K4=INT(V1*(A2-A1)/DV)+1
       DA2=(A2-A1)/K4
       K=1
       J1=0
       J2=0
       U(1)=V1*COS(A1)
       V(1)=V1*SIN(A1)
       KK1=K1+1
       KK2=K1+K2+1
       KK3=K1+K2+K3+1
       KK4=K1+K2+K3+K4+1
   10  K=K+1
       K1=K-1
       IF(KK1-K) 13,12,12
   12  U(K)=U(K1)+DV1*COS(A1)
```

```
      V(K)=V(K1)+DV1*SIN(A1)
      GO TO 10
13    IF(KK2-K) 15,14,14
14    J1=J1+1
      U(K)=V2*COS(A1+J1*DA1)
      V(K)=V2*SIN(A1+J1*DA1)
      GO TO 10
15    IF(KK3-K) 17,16,16
16    U(K)=U(K1)-DV3*COS(A2)
      V(K)=V(K1)-DV3*SIN(A2)
      GO TO 10
17    IF(KK4-K) 19,18,18
18    J2=J2+1
      U(K)=V1*COS(A2-J2*DA2)
      V(K)=V1*SIN(A2-J2*DA2)
      GO TO 10
19    M=K-1
      RETURN
      END
      SUBROUTINE BOP(T,M,N,L)
      COMMON/BLOK1/XE(100),YE(100)
      COMMON/BLOK2/U(100),V(100)
      COMMON/BLOK3/XU(100),YU(100)
      COMMON/BLOK5/X1(3000),Y1(3000),Z(3000),TA(50)
      MN=M*N
      DO 10 I=1,N
      DO 10 J=1,M
      K=M*(I-1)+J
      X1(K)=XE(I)+T*U(J)
10    Y1(K)=YE(I)+T*V(J)
      DO 15 I=1,30
15    TA(I)=TAN(0.20944*(I-1))
      DO 30 I=1,30
      IT=1
      TH=0.20944*(I-1)
      IF((TH.GT.1.5708).AND.(TH.LT.4.7124)) IT=-1
      DO 20 J=1,MN
20    Z(J)=IT*(Y1(J)-TA(I)*X1(J))
25    I0=1
      M1=MN-1
      DO 60 J=1,M1
      J1=J+1
      IF(Z(J1)-Z(I0)) 50,50,60
50    I0=J1
60    CONTINUE
      XU(I)=X1(I0)
30    YU(I)=Y1(I0)
      L=30
```

```
      DO 81 I=1,M
      PRINT 80, U(I),V(I)
   80 FORMAT(2X,F10.3,3X,F10.3)
   81 CONTINUE
        RETURN
       END
        SUBROUTINE UPDA3(A1,A2)
        COMMON/BLOK2/U(100),V(100)
        COMMON/BLOK3/XU(100),YU(100)
        COMMON/BLOK4/X0,Y0,R1,R2,S0,V1,V2,T,P
        CALL  BOS(A1,A2,V1,V2,T,P,M)
        DO 10 I=1,M
        XU(I)=X0+U(I)*T
   10   YU(I)=Y0+V(I)*T
       DO 36 I=1,M
       PRINT 35,XU(I),YU(I)
   35 FORMAT(5X,F10.3,5X,F10.3)
   36 CONTINUE
        RETURN
       END
        SUBROUTINE UPDA4(A1,A2)
       COMMON/BLOK1/XE(100),YE(100)
        COMMON/BLOK3/XU(100),YU(100)
        COMMON/BLOK4/X0,Y0,R1,R2,S0,V1,V2,T,P
        CALL BOE(X0,Y0,R1,R2,P,S0,N)
        VX=0.5*(V1+V2)*COS(0.5*(A1+A2))
        VY=0.5*(V1+V2)*SIN(0.5*(A1+A2))
        DO 10 I=1,N
        XU(I)=XE(I)+VX*T
   10   YU(I)=YE(I)+VY*T
       DO 36 I=1,N
       PRINT 35,XU(I),YU(I)
   35 FORMAT(5X,F10.3,5X,F10.3)
   36 CONTINUE
        RETURN
       END
```

APPENDIX B

MATHEMATICAL BASIS

This appendix shows that the update algorithms yields the updated confidence region with an accuracy dependent upon the number of points used to describe the boundaries of the constraint sets P and V in position and velocity space. For any set S, let ∂S denote the boundary of the set and let $\theta(\underline{x},\underline{v}) = \underline{x} + t\underline{v}$ denote the update map. Then

$$P_u = \{\theta(x,v) \mid x \epsilon P, v \epsilon V\} = \theta(P,V).$$

The following elementary results are needed:

<u>Proposition 1</u>: If the sets P and V are convex, then so is P_u.

<u>Proposition 2</u>: The boundary of P_u is a subset of the update of the boundaries of P and V. That is, $\partial P_u \subset \theta(\partial P, \partial V)$.

To prove Proposition 1, let $\underline{y}_1, \underline{y}_2 \epsilon P_u$.

Then

$$\underline{y}_1 = \underline{x}_1 + t\underline{v}_1$$

$$\underline{y}_2 = \underline{x}_2 + t\underline{v}_2$$

with

$$\underline{x}_1, \underline{x}_2 \in P$$

$$\underline{v}_1, \underline{v}_2 \in V$$

For $$0 \leq \lambda \leq 1$$

$$\lambda \underline{y}_1 + (1-\lambda)\underline{y}_2 = \theta(\lambda \underline{x}_1 + (1-\lambda)\underline{x}_2, \lambda \underline{v}_1 + (1-\lambda)\underline{v}_2).$$

But

$$\lambda \underline{x}_1 + (1-\lambda)\underline{x}_2 \in P$$

$$\lambda \underline{v}_1 + (1-\lambda)\underline{v}_2 \in V.$$

Thus

$$\lambda \underline{y}_1 + (1-\lambda)\underline{y}_2 \in P_u .$$

To prove Proposition 2, suppose $\underline{y}_o = \underline{x}_o + t\underline{v}_o \in \partial P_u$. It will be shown that if either $\underline{x}_o \notin \partial P$ or $\underline{v}_o \notin \partial V$, a contradiction is reached. Suppose $\underline{x}_o \notin \partial P$. Then there exists an open neighborhood $N_{\underline{x}_o}$ about $\underline{x}_o$ with $N_{\underline{x}_o} \subset P$. The restriction $\theta | N_{\underline{x}_o} \times \underline{v}_o$ of θ to the set $N_{\underline{x}_o} \times \underline{v}_o$ is a homeomorphism and so maps interior points of $N_{\underline{x}_o} \times \underline{v}_o$ into interior points. This implies that $\underline{x}_o + t\underline{v}_o$ is an interior point of P_u which is a contradiction. Likewise the assumption that $\underline{v}_o \notin \partial V$

leads to a contradiction. Thus $\underline{x}_0 \in \partial P$ and $\underline{v}_0 \in \partial V$.

The computer algorithms which executes the update operation proceeds as follows:

(a) Approximate the boundaries ∂P and ∂V of P and V by discrete sets $(\partial P)_0$ and $(\partial V)_0$ consisting of 30 equally spaced points.

(b) Determine the set $\theta((\partial P)_0,(\partial V)_0)$.

(c) Given a finite set $S = \theta((\partial P)_0,(\partial V)_0)$ of points, identify those points which are boundary points of the convex hull of S.

Propositions 1 and 2 assert that all boundary points of the convex set P_u are contained in the set $\theta(\partial P,\partial V)$. Since $\theta((\partial P)_0,(\partial V)_0)$ approximates the set $\theta(\partial P,\partial V)$, step (c) applied to $\theta((\partial P)_0,(\partial V)_0)$ yields an approximation to the boundary of P_u. Step (c) is discussed in Appendix C, which follows.

APPENDIX C

ALGORITHMS USED

This appendix contains a general description of two of the algorithms used in the update transformation. The first algorithm yields the boundary points of the convex hull associated with a finite set of points and may have some general interest. The second algorithm which is discussed decomposes a non-convex set in velocity space into a union of almost convex sets and is more specialized in its application.

For any set S in the plane let $C(S)$ denote the convex hull of S. The algorithm used to identify the boundary points $\partial C(S)$ of $C(S)$ can be understood by referring to Figure C1.

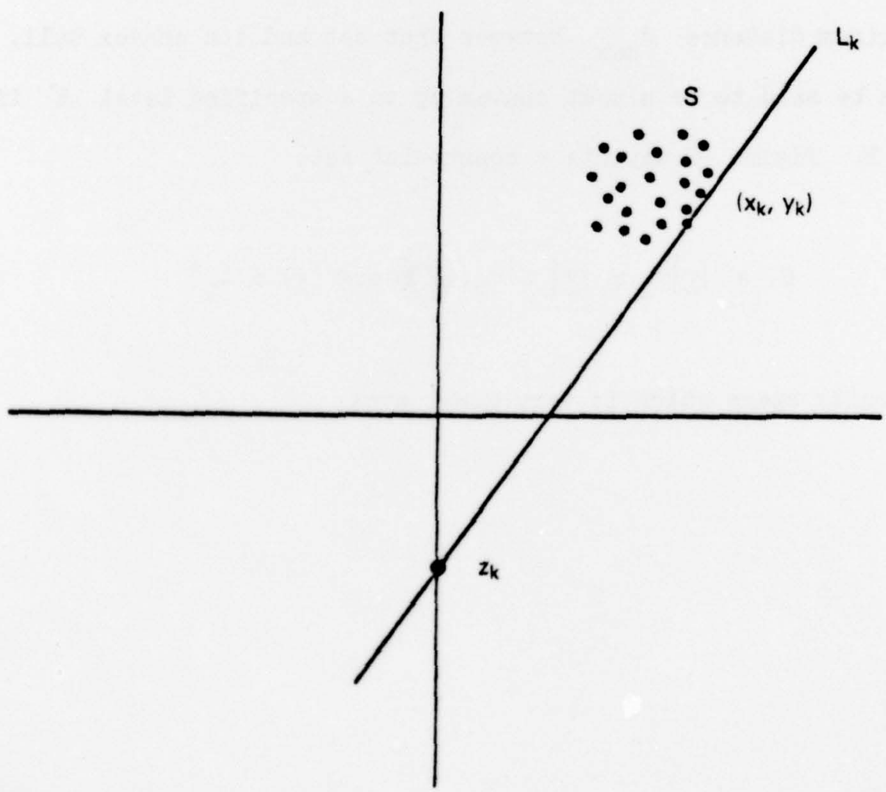


Fig. C1 — Identification of the boundary points of C(S)

The special form of the algorithm which has been programmed yields at most 30 distinct points of $\partial C(S)$. Let $\theta_k = \frac{2\pi}{30}(k-1)$ and let L_k denote a line having slope $m_k = \tan\theta_k$. Suppose $m_k > 0$. The set of all y-axis intercepts associated with the set of lines L_k (k fixed) which pass through all the point of S is determined. That point (x_k, y_k) of S for which the y-axis intercept z_k (Figure C-1) is minimized is the k^{th} point of $\partial C(S)$. Now increase k by one and repeat. If $m_k < 0$, the k^{th} point is obtained by maximizing the set of all y-axis intercepts. Note that for all k, θ_k never takes on the value $\frac{\pi}{2}$ or $\frac{3\pi}{2}$. The set $\{(x_1,y_1), (x_2,y_2), \text{---} (x_{30},y_{30})\}$ contains at most 30 distinct points of $\partial C(S)$.

An almost convex set in velocity space can be defined relative to the maximum distance d_{max} between that set and its convex hull. A set can be said to be almost convex up to a specified level E if $d_{max} \leq E$. Figure C2 depicts a constraint set,

$$V = \{\underline{v} \mid v_1 \leq |\underline{v}| \leq v_2, A_1 \leq \arg(\underline{v}) \leq A_2\}$$

in velocity space which is very non-convex.

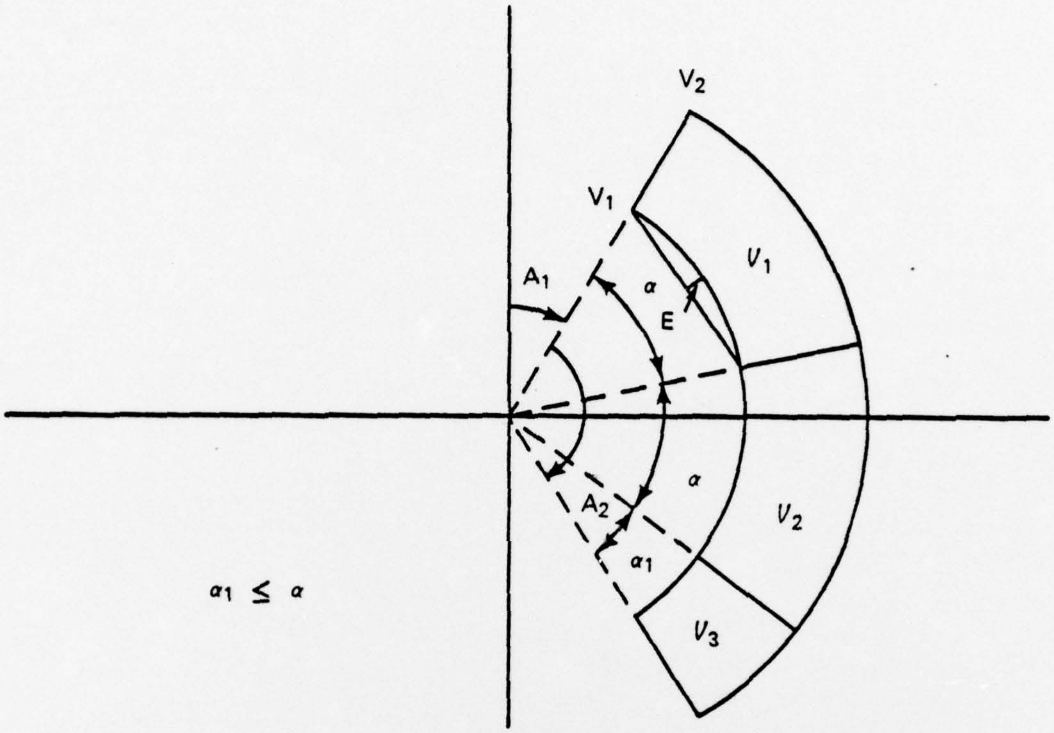


Figure C2

Decomposition of a Non Convex Set into a Union of Almost Convex Sets

However, by choosing the angle α (Figure C-2) small enough, the set V decomposes into a union

$$V = V_1 \cup V_2 \cup V_3$$

where each of the sets V_i are almost convex up to level E. The angle α which determines the size of the sets V_i is related to E and v_1 by

$$\alpha = 2 \cdot \text{Arccos}\left(1 - \frac{E}{v_1}\right).$$